In the Claims:

Please amend the claims as follows:

1. (Currently Amended) A programmable processor for executing a plurality of programs, said programmable processor comprising:

an execution pipeline having ~~a~~ an average pipeline latency of one instruction per cycle; and

an interleaver for interleaving instructions from said plurality of programs and providing said instructions to said pipeline for execution such that the number of said plurality of programs that are interleaved is greater than or equal to the depth of the pipeline.

2. (Previously Presented) The processor of claim 1 or claim 23 wherein said pipeline has a datapath with a depth equal to said number of programs.

3. (Previously Presented) The processor of claim 1 wherein a next instruction from one of said plurality of programs is not provided to said pipeline until a previous instruction of said one of said plurality of programs has completed.

4. (Previously Presented) The processor of claim 1 or claim 23 wherein each program of said plurality of programs is independent of the other of said plurality of programs.

5. (Previously Presented) The processor of claim 1 or claim 23 further including an output buffer for storing out of order data output.

6. (Previously Presented) The processor of claim 1 or claim 23 further including one or more of a register copy, program counter, and program counter stack provided for each of said plurality of programs.

7. (Previously Presented) The processor of claim 6 wherein one or more of control and computing resources, instructions, instruction memory, data paths, data memory, and caches are shared by said plurality of programs.

8. (Previously Presented) The processor of claim 1 or claim 23 wherein said processor executes SIMD vector instructions of vector length N and executes in parallel a plurality of instructions having SIMD vector lengths that sum up to N.

9. (Previously Presented) The processor of claim 1 or claim 23 wherein said instructions comprise load instructions for loading data from a data memory.

10. (Previously Presented) The processor of claim 1 or claim 23 wherein said instructions comprise store instructions for storing data in a memory.

11. (Previously Presented) The processor of claim 9 wherein said data memory comprises a cache.

12. (Previously Presented) The processor of claim 9 wherein address space of said data memory comprises a frame buffer unit.

13. (Previously Presented) The processor of claim 9 wherein address space of said data memory comprises a texture memory unit.

14. (Currently Amended) A method of executing instructions from a plurality of programs comprising:
identifying N programs of said plurality of programs;
interleaving instructions from said N programs in a processor pipeline wherein said pipeline has an average latency of one instruction per cycle; and

executing said instructions such that a first instruction from one of said N programs is completed before beginning execution of a second instruction of said one of said N programs wherein no no-op is inserted into the pipeline for the purpose of ensuring that said first instruction is completed before beginning execution of said second instruction.

15. (Previously Presented) The method of claim 14 or claim 24 further including the step of assigning a program counter to each of said N programs.

16. (Previously Presented) The method of claim 14 or claim 24 further including the step of assigning a register to each of said N programs.

17. (Previously Presented) The method of claim 14 or claim 24 wherein said graphics processing execution pipeline has a depth of N.

18. (Previously Presented) The method of claim 14 or claim 24 wherein said processor executes SIMD vector instructions of vector length N and executes in parallel a plurality of instructions having SIMD vector lengths that sum up to N.

23. (Currently Amended) A programmable processor for executing a plurality of programs, said programmable processor comprising:
an execution pipeline having ~~a~~ an average pipeline latency of one instruction per cycle; and
an interleaver for interleaving instructions from said plurality of programs and providing said instructions to said pipeline for execution wherein a next instruction from one of said plurality of programs is not provided to said pipeline until a previous instruction of said one of said plurality of programs has completed and wherein no no-op is inserted into the pipeline for the purpose of ensuring that said next instruction is not provided to said pipeline until said previous instruction has completed.

24. (Currently Amended) A method of gathering ~~executing~~ instructions from a plurality of programs comprising:

identifying N programs of said plurality of programs wherein N is greater than or equal to the depth of a processor pipeline;
interleaving instructions from said N programs in said processor pipeline; and
executing said instructions.

25. (New) A programmable processor for executing a plurality of programs, said programmable processor comprising:
a target program counter coupled to a plurality of program counters;
each of said plurality of program counters coupled to an instruction memory;
instructions from said instruction memory coupled to an instruction decode;
said decode coupled to a plurality of registers;
each of said plurality of registers coupled to an operand route;
said operand route coupled to an arithmetic datapath;
said datapath and an output of a data memory coupled to a result route; and
an output of said result route fed back to each of said plurality of registers.

26. (New) The programmable processor of claim 25 wherein said plurality of program counters is equal to said plurality of programs to be interleaved.

27. (New) The programmable processor of claim 25 wherein said plurality of registers is equal to said plurality of programs to be interleaved.

28. (New) The programmable processor of claim 25 wherein said plurality of registers is more than said plurality of programs to be interleaved.

29. (New) The programmable processor of claim 25 wherein said instruction memory is larger than needed to hold said plurality of programs.

30. (New) The programmable processor of claim 25 wherein said data memory is larger than needed to hold a data set accessed by each of said plurality of programs.

31. (New) The programmable processor of claim 25 wherein each of said plurality of registers is double buffered and contains twice as many copies of registers as said plurality of programs.

32. (New) A method of executing one or more complex or compound instructions from a plurality of programs, comprising:
implementing said instructions in one or more pipelined units wherein each of said instructions is issued to said one or more units in each cycle.

33. (New) A method of executing one or more instructions from a plurality of programs, comprising:
assigning a first output register slot to a first of said plurality of programs;
executing said one or more instructions of said first program until said first program is completed;
loading output of said first program into its reserved space when said first program is completed;
checking to see if all of said plurality of programs are completed;
checking to see if a second output register slot is available to assign to a second program from said plurality of programs when said first program is completed;
checking to see if one or more instructions are available when at least one of said plurality of programs is not completed; and
placing an no-op when no more instructions are available or said second output register slot is not available.

RESPONSE

The Applicant has found a few typographical errors in the specification and would like to correct them and states that the amendments made to the specification do not raise any new matter.

The Examiner has rejected claims 1-18, 23, and 24 in view of new grounds. The Applicant has amended claims 1, 14, 23, and 24, and added new claims 25-33.

Claim Rejections based on 35 USC 103

Claims 1, 2, 4-7, 9-11, and 24 are rejected under 35 USC 103(a) as being unpatentable over US Patent 5,768,594 to Blelloch et. al. in view of US Patent 6,493,820 B2 to Akkary et. al. Applicant respectfully disagrees because neither Blelloch nor Akkary teach, suggest, or describe a programmable processor that has an average latency of one instruction per cycle.

Claim 8 is rejected under 35 USC 103(a) as being unpatentable over US Patent No. 5,768,594 to Blelloch et. al. in view of US Patent No. 6,493,820 to Akkary et. al. as applied to claim 1 above, and further in view of US Patent No. 5,961,628 to Nguyen et. al. Applicant respectfully disagrees because independent claim 1 as amended is not taught, suggested, or described by Blelloch, and further claim 8 is dependant on a now allowable independent claim.

Claims 12 and 13 are rejected under 35 USC 103(a) as being unpatentable over US Patent No. 5,768,594 to Blelloch et. al. in view of US Patent No. 6,493,820 B2 to Akkary et. al. as applied to claim 1 above, and further in view of US Patent No. 5,973,705 to Narayanaswami. Applicant respectfully disagrees because independent claim 1 as amended is not taught, suggested, or described by Blelloch, and further claims 12 and 13 are dependant on a now allowable independent claim.

Claims 3, 14-18, and 23 are rejected under 35 USC 103(a) as being unpatentable over US Patent No. 5,768,594 to Blelloch et. al. in view of US Patent No. 6,493,820 B2 to Akkary et. al. as applied to claim 1 above, and further in view of US Patent No. 6,209,083 B1 to Naini et. al. Applicant respectfully disagrees because independent claims 1 and 14 as amended are not taught, suggested, or described by Blelloch, and further claims 3, and 15-18 are dependant on a now allowable independent claim.